Normalizing early English letters to Present-day English spelling

Hämäläinen, Mika

The Association for Computational Linguistics
2018

Hämäläinen , M , Säily , T , Rueter , J , Tiedemann , J & Mäkelä , E 2018 , Normalizing early English letters to Present-day English spelling . in B Alex , S Degaetano-Ortlieb , A Feldman , A Kazantseva , N Reiter & S Szpakowicz (eds) , Proceedings of the 2nd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature . ACL Anthology , no. W18-45 , The Association for Computational Linguistics , Stroudsburg, PA , pp. 87-96 , Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature , Santa Fe , New Mexico , United States , 25/08/2018 . < http://aclweb.org/anthology/W18-4510 >

http://hdl.handle.net/10138/308739

Normalizing Early English Letters to Present-day English Spelling

Mika Hämäläinen, Tanja Säily, Jack Rueter, Jörg Tiedemann and Eetu Mäkelä
Department of Digital Humanities
University of Helsinki
firstname.lastname@helsinki.fi

Abstract

This paper presents multiple methods for normalizing the most deviant and infrequent historical spellings in a corpus consisting of personal correspondence from the 15th to the 19th century. The methods include machine translation (neural and statistical), edit distance and rule-based FST. Different normalization methods are compared and evaluated. All of the methods have their own strengths in word normalization. This calls for finding ways of combining the results from these methods to leverage their individual strengths.

1 Introduction

Working with historical texts is a challenging task for NLP. Whereas many off-the-shelf tools and libraries are available for the modern written standard of a majority language, these are not directly applicable to historical data. This is due to a wide variety in how words are spelled.

The problem of non-standard spelling is not an easy one to tackle because there is a multitude of reasons affecting orthography. Not only is there variation in spelling norms in different centuries but also individual variation affected by the dialect background of the writers, their command of the written norm, and the level of formality of the text among other factors. Orthography has also been influenced by editorial conventions at different stages of bringing the historical texts into a digital format.

The goal of this paper is to propose and compare methods of normalizing historical English automatically in the context of our CEEC corpus (*Corpora of Early English Correspondence*) (Nevalainen et al., 1998 2006), which consists of letters ranging from the 15th to the 19th century. The corpus consists of personal correspondence, which exhibits a great variety of non-standard spellings. The word *about* alone, for example, has over ten different spelling variants in our corpus such as *aboutt*, *aboute*, *abowt*, *abovt* and so on.

We will first normalize as much as possible of the CEEC with existing tools and methods. This will leave us with the most difficult cases of variant spelling to be dealt with using the methods we are presenting and evaluating in this paper. In other words, it is not in our interest to try to normalize something we can get the normalizations for already, but rather focus our efforts on the most deviant spelling forms.

Normalizing the CEEC will allow us to conduct more NLP research on the data in the future by using tools and libraries available for Present-day English. The primary motivation for our normalization efforts is studying neologisms and sociolinguistic variation in the letters. This means that we need to be able to normalize even the most difficult and infrequent spellings in our corpus.

2 Related Work

In the past, normalization of old texts has received some attention as an NLP task. There are ready-made tools available for normalization such as VARD2 (Baron and Rayson, 2008) and MorphAdorner (Burns, 2013). These tools, however, are not sufficient to solve the problem automatically for our corpus. Using VARD2 requires manual work and MorphAdorner does not provide enough coverage for our data.

Proceedings of Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature, pages 87–96
Santa Fe, New Mexico, USA, August 25, 2018.

Using a string similarity metric such as edit distance has been used in the past for normalization. An example of such is the automatically produced diachronic dictionary of spelling variants for German (Amoia and Martinez, 2013). In building the dictionary, the authors used Levenshtein edit distance to cluster similar words together with their modern counterparts. This was facilitated by the fact that they were dealing with recipes, which thanks to their limited vocabulary, are easy to cluster. In addition, words can be clustered based on their semantics by looking at the shared contexts of the words.

Statistical machine translation models have also been used in solving the task by training a character based translation model using known historical spellings and their modern variants as training data. In (Samardzic et al., 2015) such a model was trained for normalizing Swiss German dialects to a standard variant. A similar SMT based approach has also been used in the context of historical text in (Pettersson et al., 2013).

Normalization has also been done by using deep learning. In (Bollmann and Søgaard, 2016), normalization is presented as a character-based sequence labeling task for bi-directional LSTMs. In their method a historical character does not need to be aligned with a single modern spelling character but can be aligned with a compound of characters in the training data.

3 The Corpus and Data Sources

The primary corpus we use is the CEEC (Nevalainen et al., 1998 2006). Compiled for the purposes of historical sociolinguistics, it is a corpus of personal correspondence in English representing a time span from the 15th to the 19th century. The letters have been selected from published original-spelling editions and digitized by the compilers, who have hand-corrected the OCR. The corpus comprises c. 5 million running words in 12000 letters written by more than 1000 individuals. The authors of the letters come from varied social backgrounds as reflected by the metadata in the CEEC, which contains, e.g., the socioeconomic status, gender, age, domicile and the relationship between the writer and recipient.

3.1 Preparing the Corpus

As a first step, we convert the CEEC into Unicode, remove punctuation, excluding apostrophes and hyphens, and tokenize the letters. After this step, we attempt to lemmatize each token with NLTK (Bird et al., 2009) by using the lemmatizer based on the Princeton WordNet. After this, we try to map each lemmatized token into the *Oxford English Dictionary* (OED). By doing so, we can filter out all the words that have a modern spelling in our corpus leaving out the ones with a non-standard spelling.

Our endeavor is to map as many words with non-standard spelling to the OED as possible. The most modern parts of the CEEC, namely those from the 16th to the 18th century, have been partially normalized with VARD2 in the past. These normalizations have been tagged in the CEEC in such a way that the original non-standard form has also been retained. We use this parallel list of non-standard and standard forms to map even more words to the OED by extending these normalizations to the whole corpus.

We run a tool called MorphAdorner on our corpus to produce more normalizations automatically. This allows us to map even more words to the OED. At this point, we have normalized all of the ”easy” cases leaving out the words that have not been mapped into the OED utilizing these approaches. The number of unique unmapped word forms is 85298. The rest of the paper focuses on the normalization of these difficult word forms.

The fact that we were able to map some of the word forms to the OED has provided us with some parallel data specific to our corpus. We use the VARD2 normalizations and the MorphAdorner produced ones as a starting point for compiling a parallel corpus of non-normalized and normalized word forms. This corpus is extended with more parallel data in the next step, described in the following subsection of this paper.

3.2 Extracting Data from Other Sources

In order to normalize the old spellings in the CEEC, we use two lexicographical sources that link old spelling variants to their modern counterparts, namely the Oxford English Dictionary (OED, nd) and the

Middle English Dictionary (MED, nd), both of which we have in XML format. These two resources provide us with indispensable parallel data of non-standard and standard word forms.

We also use the British National Corpus (The BNC Consortium, 2007) as a language model for statistical machine translation. The BNC does not provide us with any parallel data since it does not have historical texts. We can, however, take the individual words out of the corpus to build a language model for the SMT system.

The OED stores alternative spellings in multiple parts of the hierarchical XML structure. To extract the non-standard and standard pairs from the OED, we use the information stored in the derivative lemmas of the lemma section as well as the forms and variant forms subsection of the inflection section, which stores only historical variants of the lemma. The latter section also contains valuable information about the centuries the word form was used in; we also store this information when we extend our initial parallel corpus with this new data. All of these sections contain variant forms that are relevant for our normalization efforts, because ultimately our goal is to map all the words in the CEEC to the OED entries.

We can get a similar historical–modern spelling parallel collection of words from the MED, the entries of which are Middle English word forms. The MED entries themselves do not contain a modern normalized version of the Middle English form, but the MED comes with a continuously developed XML[1] which contains links between some of the entries in the MED and in the OED. From this XML we can extend our parallel corpus with the historical MED headwords and their normalizations. The MED also provides us with the time period a given word form was in use, which we store as well. At this point, our parallel data consists of 183505 pairs of historical and modern spellings.

4 Methods for Normalization

In this part we present different methods for normalizing historical English words. The idea is to leverage each of their strengths by having all of them provide their own suggested normalizations. It is from this list of possibilities that the most likely normalization is picked.

4.1 VARD2 Rules

As previously mentioned, the most modern parts of the CEEC have been normalized to some extent with VARD2. As one possible normalization method, we apply these VARD2 rules to the list of non-normalized words in the CEEC. The VARD2 rules are simple replacement rules of character sequences given their position in the word. The position of a sequence can be anywhere, even at the beginning or the end of a word. All in all, we have a list of 58 unique replacement rules, such as "u $\rightarrow$ v anywhere".

We build a Finite-State Transducer (FST), using HFST (Lindén et al., 2013), based on the VARD2 rules. This allows multiple combinations of the VARD2 normalization rules to be applied for producing normalized forms. We define the LEXC file so that it consists of the lemmas from the OED. The normalization rules are defined with weighting to a spellrelax.regex file following a description in spelling (Beesley and Karttunen, 2003). The FST then applies the rules in all possible combinations to match the non-normalized forms to forms in the LEXC file. The weighting shows how many modifications were needed to reach normalization. We consider the candidate with least modifications required as the normalization candidate produced by this method.

4.2 Contextual Edit Distance

This method compares the non-normalized words to our extracted list of unique words from the BNC. First we list the possible normalization candidates by comparing the non-normalized words to the words in the BNC applying Levenshtein edit distance. The Levenshtein distance gives a score on the string similarity of two words, where each difference, such as addition or deletion of a character, increases the distance. In this way, we collect sets of normalization candidates for each non-normalized word consisting of candidate words with an edit distance score of 3 or lower.

[1]The version we use in this paper is dated 3/2018.

The list of normalization candidates is still too extensive and needs to be filtered down to fewer possibilities. This is done by looking at the shared context of the non-normalized words in the CEEC and the candidates in the BNC. In practice, for each non-normalized word and normalization candidate, we get the two words that precede and the two words that follow said word in each and every occurrence. Thus, we have built a list of contextual words together with information about their position in relation to each word, non-normalized and candidate alike. We then use these lists to further filter the normalization candidates, inspecting the intersection length of the lists for contextual words given their position. We only retain the normalization candidates exhibiting the highest number of contextually shared words with the non-normalized word.

Although the previous step narrows down the normalization candidates, we still need to filter them further. As a last step, we look at the pronunciation of the candidates and compare this with the pronunciation of the non-normalized words. Since we cannot produce an accurate pronunciation for the non-normalized words due to the fact that English orthography and pronunciation are not always clearly connected, we use Soundex[2] with the size of 6 to produce an estimated pronunciation.

4.3 Statistical Machine Translation (SMT)

Previous research has shown that SMT is a viable form of solving the problem of normalization. This is why we have decided to include an SMT based approach as a module in our system. We train a character based SMT model using the parallel data extracted earlier from the OED, MED and known normalizations in the CEEC. All the words are split into letters separated by a whitespace in order to make the SMT tool, Moses (Koehn et al., 2007), treat individual characters of a word as though they were words of a sentence. The parallel non-normalized to normalized word lists are aligned with GIZA++ (Och and Ney, 2003) as part of the machine translation process.

An SMT based system also requires a language model. Without it, the system would be more likely to produce non-words as output. As a language model, we use the list of words extracted from the BNC. Again, these words are split into characters by whitespaces. We build a 10-gram language model based on the BNC data with KenLM (Heafield et al., 2013) and use this model with Moses[3].

For tuning the model, we take a random sample of 2000 non-normalized and normalized word pairs from our parallel data set and run the tuning on that. We also tune century specific models with 2000 words from the era for the 15th and the 18th century to compare whether tuning for a given century yields better results for that century than a more generally tuned model.

4.4 Neural Machine Translation (NMT)

Neural machine translation can be used for normalization in a similar fashion to the SMT approach. We use OpenNMT (Klein et al., 2017) to train against a character based machine translation model by using the parallel data extracted earlier. For validation of the model, we use the same 2000 word parallel list as we did for the general SMT model.

For NMT, we train two different models for 13 epochs. The first model gets the same input as SMT, i.e. a parallel list of word forms. The second model has a specific input for the centuries in which certain non-normalized forms were used together with the actual word forms. This is done simply by appending the year of the century before each historical form in the data set. For some of the word forms, we do not have the information of the time period in which they were used. In such cases, we include the word forms without a year label.

For the model trained with year labels, we also add the year labels to the list of non-normalized word forms we input to the system for it to translate them. These years can be recovered from the letter metadata in the CEEC.

Additionally, we take the trained year aware model and continue training it separately for the 15th and the 18th century by feeding in the parallel data of only those centuries for an additional 10 epochs. For validation, we use a random sample of 2000 word pairs for the time periods. We do this to see whether by

[2]We use the implementation of the Fuzzy Python package (https://pypi.org/project/Fuzzy/).

[3]Note: Moses has to be built with a separate flag to allow language models that are the size of a 10-gram model.

continuing the training, the model can learn a more accurate normalization model specific to one century while still taking advantage of the normalizations for all the centuries.

4.5 Combining the Approaches

All the previously described approaches result in their own normalization candidates. In order to take advantage of them all, we need to pick out a normalization from all of the possibilities. Since we have an SMT and an NMT approach in use, some of the normalization candidates might be non-words as the machine translations can output words that are not part of the English language. First, we look up the normalization candidates in the OED. If they are not English words according to the OED, we ignore them and do not consider them to be selected.

For the remaining normalization candidates, we try two different approaches to picking out the correct one. The first one is a simple voting mechanism: the more approaches result in the same normalization, the more likely it is that correct normalization has been found. In the case of a tie in the voting, we pick a candidate at random out of the top ones.

The other approach is finding the normalization best fitting the context. We do this by training a first order Markov chain on the BNC. The chain learns the probability at which a given word follows another in a sentence. We can use this probability to pick the normalization that is the most likely to fit a given context.

When using the chain to pick out the best normalization, we look at a context window of 4 tokens on each side of the word to be normalized. Then we check whether all the words in the context are normalized, and replace them with their normalizations, if they are known. If, however, there are words that cannot be normalized in the context window, we will also look at the normalization candidates produced by our methods for those words. Then we count the probabilities for the chain by all the possible different combinations of normalizations and pick the likeliest one. For transitions of words that are not in the model, we add a probability of 1 over the number of possible transitions from that state.

5 Results and Evaluation

To evaluate the results produced by each individual method and the overall performance of combining them, we prepare three gold standards (Säily, 2018) by normalizing a random sample of 100 words by hand out of the non-normalized words in the CEEC for each of them. The first set has words randomly picked from all centuries and the other two consist of randomly picked 15th and 18th century words respectively. These normalizations are made by a linguist who is familiar with historical English texts.

5.1 Individual Methods

We compare the outputs of each of the methods to the gold standards presented earlier both directly and using NLTK WordNet Lemmatizer on the forms to see if the lemmas are the same. Since our ultimate goal is to map the words to the OED, we are more interested in matching to the right lemma rather than the exact inflectional form of the word. The results are shown in Table 1, where the columns represent the three different gold standards and the rows accuracies of each method. The last row shows the percentage of words that were correctly normalized by at least one method. This represents the upper boundary of accuracy that our system can achieve when selecting the best suitable normalization.

The first three NMT models are the ones with the century information fed into them during the training and the "NMT no years" is the one that received only words without their centuries as input. It is clear that the NMT model without the information about the century outperforms the three century specific ones.

Figure 1 shows the overlap of the correct normalizations produced by each method in a Venn diagram in the case of the general test set. The leftmost diagram shows a combination of the different SMT and century aware NMT models, while the differences in between the SMT and century aware NMT models are shown in the other two diagrams. The number in the bottom right corner shows the number of normalizations that were not produced correctly by any of the methods in the diagram.

Model	Generic	15th century	18th century
NMT	28%	43%	14%
NMT 18th	28%	43%	15%
NMT 15th	28%	43%	15%
NMT no years	46%	55%	25%
SMT	32%	31%	28%
SMT 18th	16%	14%	19%
SMT 15th	32%	28%	31%
Edit distance	31%	31%	31%
VARD2 rules	15%	10%	8%
At least one correct	*67%*	*71%*	*52%*

Table 1: Accuracy of each method

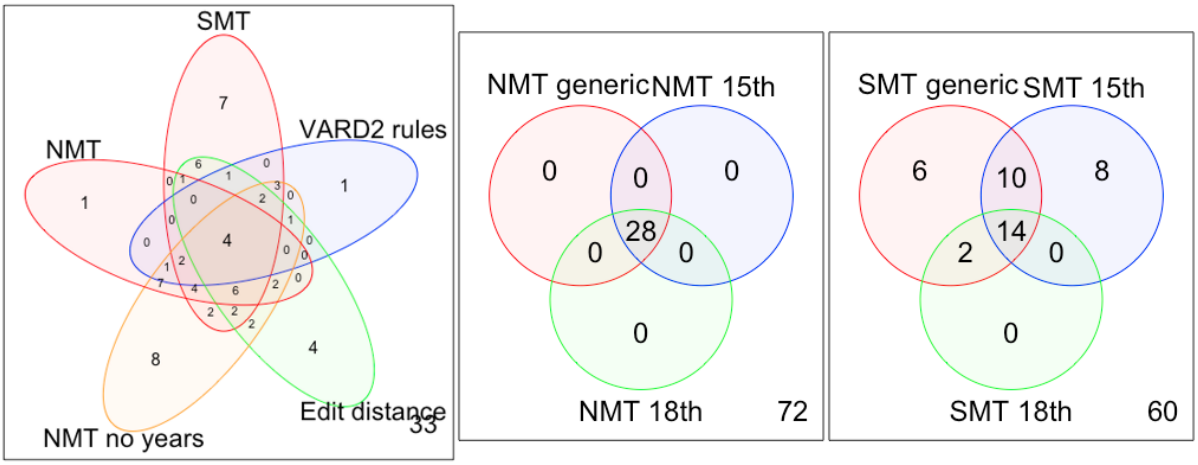

Figure 1: Overlap of correct normalizations in the general test set

We can see from Figure 1 that all of the methods have been able to correctly normalize words the other methods have not. There is also a reasonable overlap in the correct normalizations of multiple methods. This means that while a voting approach in picking the correct normalization out can work with a high precision, its recall will be lower due to the high amount of correct normalizations produced by only one method.

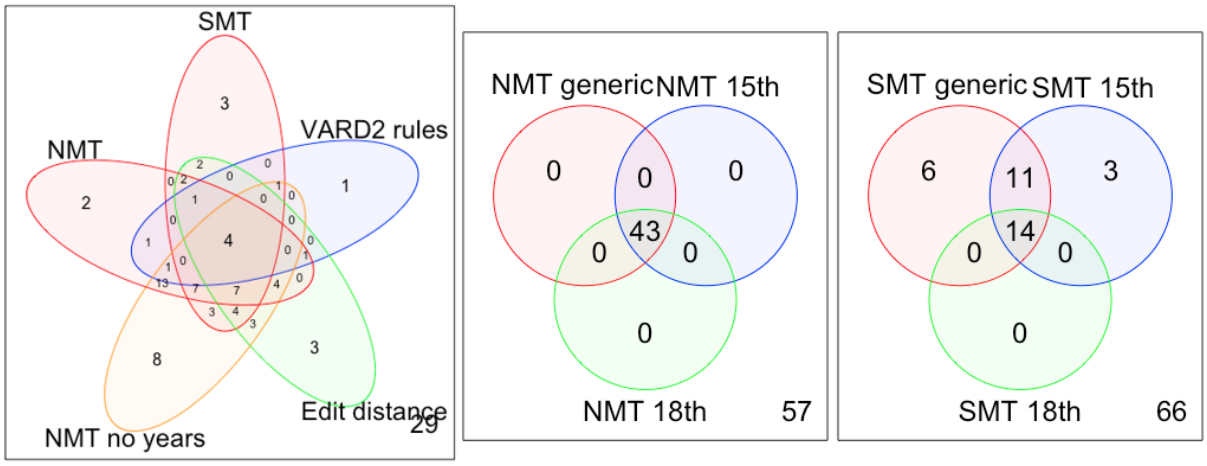

Figure 2: Overlap of correct normalizations in the 15th century test set

As for the century specific machine translation models, in the general dataset, all of the century aware NMT models produce the very same normalizations. In the case of the SMT models, both the generic one and the 15th century one find correct normalizations of their own, while the 18th century model fails to produce additional correct normalizations not produced by the other two.

Figure 2 depicts the overlap of correct normalizations in the 15th century test set. Again, all of the methods have managed to produce correct normalizations not captured by the other methods. The best one at producing correct normalizations not produced by the others is the NMT model that has not been trained with the century information, with its 8 unique correct normalizations.

Despite the high accuracy of the century aware NMT models, we can see in Figure 2 that no specialization has occurred in the 15th century NMT model to improve work for this century. In fact, all of the century-aware NMT models still produce the exact same normalizations.

As for the SMT models, we can see that the 15th century model is able to correctly normalize 3 words the other SMT models have not. What is more interesting is that despite the specialization for this century, the 15th century model has failed to correctly normalize 6 words the generic model was able to normalize.

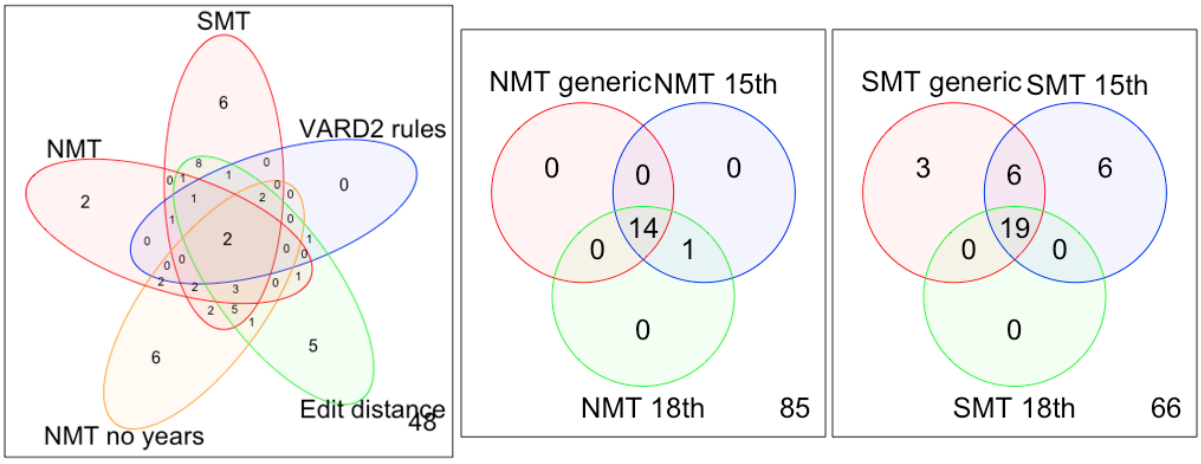

Figure 3: Overlap of correct normalizations in the 18th century test set

Finally, Figure 3 represents the overlapping correct normalizations for the 18th century. This time around, all methods except for the VARD2 rules have been able to produce correct normalizations of their own. The century aware NMTs are the weakest ones in this respect.

Interestingly, both of the century specific NMT models have been able to find the same additional correct normalization where the generic century-aware model has failed. Otherwise the normalizations produced by the models are the same.

Curiously, in the case of the SMT models, the one tuned for the 18th century seems to perform the worst, while the generic one and the 15th one are both able to produce unique normalizations of their own. In addition these two models both find the same 6 correct normalizations that the model tuned for the 18th century does not.

All in all, looking at the evaluations, it is contextual edit distance that seems to be the most consistent at finding the same amount of correct normalizations in each test set and always finding some unique ones. The same applies to the generic SMT model, which is slightly less consistent than the contextual edit distance. The VARD2 rewrite rules, unsurprisingly, perform the worst. And all the NMT models preform very differently in different test sets.

5.2 Picking the Best Normalization

In this part we present the results of the two ways for picking out the likeliest normalization from the normalization candidates produced using the different methods. These are voting and using the Markov chain trained earlier to pick out the likeliest normalization. We run the voting weighted with the accu-

racies of the individual methods so that the methods that have a higher accuracy have a higher vote. In addition, we run the voting unweighted so that every method has an equal vote.

	Generic	15th century	18th century
Voting no weights	41%	49%	23%
Voting weighted	45%	56%	22%
Markov chain	36%	40%	24%

Table 2: Accuracy of different ways to pick out the correct normalization

Table 2 shows that the weighted voting gives the highest results in picking the correct normalization candidate out of the possible ones in almost all the test sets. The Markov chain only outperforms it in the 18th century.

The reason for the poor performance of the Markov chain model is probably that we have narrowed this task down to the historical forms that are not trivially normalizable. This means that we are dealing with non-standard forms that occur in the CEEC only a handful of times in a context where other non-normalized word forms co-occur. This makes it difficult for a statistical model to pick out the most suitable normalization with only limited information available on the surrounding context.

6 Discussion and Future Work

The machine translation models were trained with the idea that the spelling variation follows similar tendencies within a century. Looking at the results, especially the ones obtained with SMT, it seems that the century is not necessarily a good enough variable to look at. While variation might not always be similar within a century, in the future we should look at the other variables recorded in the CEEC, such as the social class of the writer and the collection in which the letter has been published. Since this information is CEEC specific, however, having enough training data to train normalization for example for a certain social class becomes an issue.

In addition, we need to find a better way of making the NMT models specialize in a given category of variance, let it be the century as in the trials presented in this paper, or the social class of the author of the letter. Our results show that continuing the training with only the data for a particular century with a validation dataset of that century is not enough for the model to perform any better in its supposed century of specialization.

For NMT, feeding in century labels along with the historical word forms seems to even be harmful to the accuracy of the system. Finding better alternative ways to include additional information in the NMT model that will not make the accuracy decline is also a task for future research.

As for combining the results of the approaches, there is definitely room for improvement. Currently, the best performing method is weighted voting, but based on the Venn diagrams for the overlap of the correct normalizations by different methods, we can conclude that the voting approach will never reach the upper boundary of possible accuracy for the entire system: the number of correct normalizations that are unique to only one of the methods is too high, in which case voting will not suffice for selecting them. This is especially the situation if a method with a lower weight in its vote is the only one with the correct normalization.

In this paper, we have taken a naive approach thinking that two identical historical forms always normalize to the same standard form. This is true to a great extent in our case where we are dealing with the least frequent deviant forms, but there are still cases even in our set of non-normalized words where two identical historical spellings normalize into different modern English words. An example of such is *querry*, which is either *equerry* or *query* depending on the letter of appearance.

The ultimate goal of our normalization efforts is to facilitate the next step in our research, which is finding neologisms automatically in the CEEC. This is done by comparing the year of the letter where a neologism candidate is used with the earliest attestation recorded in the OED. This, of course, requires that we are able to map every single word in the CEEC to the OED. Due to the nature of neologisms, we are interested even in the least frequent words in the corpus.

7 Conclusions

In this paper, we have presented different approaches to normalized deviant infrequent historical spellings in the CEEC corpus. While all of the individual approaches have strengths of their own in terms of their facilitating the normalization of spelling variants other methods cannot, the single best individual method seems to be the NMT method without additional century labels.

Century has proven to be a poor variable to base tuning of the models on in the case of our corpus. This must be due to the fact that most of the spelling variation is due to other factors than the century of writing such as dialect background, social class and individual spelling mistakes.

Based on the findings presented in this paper, we have multiple different directions to continue our normalization efforts, all of which require further research and thus fall outside the scope of this paper.

Acknowledgements

This work was supported by the Academy of Finland grant 293009.

References

Marilisa Amoia and Jose Manuel Martinez. 2013. Using comparable collections of historical texts for building a diachronic dictionary for spelling normalization. In *Proceedings of the 7th workshop on language technology for cultural heritage, social sciences, and humanities*, pages 84–89.

Alistair Baron and Paul Rayson. 2008. VARD2: a tool for dealing with spelling variation in historical corpora. In *Postgraduate Conference in Corpus Linguistics*. Aston University, Birmingham.

Kenneth R. Beesley and Lauri Karttunen, 2003. *Finite-State Morphology*, pages 451–454. Stanford, CA: CSLI Publications.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

Marcel Bollmann and Anders Søgaard. 2016. Improving historical spelling normalization with bi-directional LSTMs and multi-task learning. *CoRR*, abs/1610.07844.

Philip R Burns. 2013. MorphAdorner v2: A Java library for the morphological adornment of English language texts. *Northwestern University, Evanston, IL.*

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proc. ACL.*

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL on interactive poster and demonstration sessions*, pages 177–180. Association for Computational Linguistics.

Krister Lindén, Erik Axelson, Senka Drobac, Sam Hardwick, Juha Kuokkala, Jyrki Niemi, Tommi A. Pirinen, and Miikka Silfverberg. 2013. HFST — A System for Creating NLP Tools. In Cerstin Mahlow and Michael Piotrowski, editors, *Systems and Frameworks for Computational Morphology*, pages 53–71, Berlin, Heidelberg. Springer Berlin Heidelberg.

MED. n.d. Middle English Dictionary. University of Michigan. https://quod.lib.umich.edu/m/med/.

Terttu Nevalainen, Helena Raumolin-Brunberg, Jukka Keränen, Minna Nevala, Arja Nurmi, Minna Palander-Collin, Samuli Kaislaniemi, Mikko Laitinen, Tanja Säily, and Anni Sairio. 1998–2006. CEEC, Corpora of Early English Correspondence. Department of Modern Languages, University of Helsinki. http://www.helsinki.fi/varieng/CoRD/corpora/CEEC/.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

OED. n.d. OED Online. Oxford University Press. http://www.oed.com/.

Eva Pettersson, Beáta Megyesi, and Jörg Tiedemann. 2013. An SMT approach to automatic annotation of historical text. In *Proceedings of the workshop on computational historical linguistics at NODALIDA 2013; May 22-24; 2013; Oslo; Norway. NEALT Proceedings Series 18*, number 087, pages 54–69. Linköping University Electronic Press.

Tanja Säily. 2018. Test set for Normalization of Historical English in CEEC, June. https://doi.org/10.5281/zenodo.1300332.

Tanja Samardzic, Yves Scherrer, and Elvira Glaser, 2015. *Normalising orthographic and dialectal variants for the automatic processing of Swiss German*. Proceedings of the 7th Language and Technology Conference. ID: unige:82397.

The BNC Consortium. 2007. The British National Corpus, version 3 (BNC XML Edition). Distributed by Bodleian Libraries, University of Oxford, on behalf of the BNC Consortium. http://www.natcorp.ox.ac.uk/.